Fast Blur Detection and Parametric Deconvolution of Retinal Fundus Images

Bryan M. Williams[1], Baidaa Al-Bander[2], Harry Pratt[1], Samuel Lawman[2], Yitian Zhao[4,5], Yalin Zheng[1,3], and Yaochun Shen[2]

[1] Dept. of Eye and Vision Science, University of Liverpool, UK, [bryan,yzheng]@liverpool.ac.uk
[2] Dept. of Electrical Engineering & Electronics, University of Liverpool, UK
[3] St. Paul's Eye Unit, Royal Liverpool University Hospital, Liverpool, UK
[4] Chinese Academy of Sciences, Ningbo Institute of Material Technology and Engineering, Ningbo, China
[5] School of Optics and Electronics, Beijing Institute of Technology, Beijing, China

Abstract. Blur is a significant problem in medical imaging which can hinder diagnosis and prevent further automated or manual processing. The problem of restoring an image from blur degradation remains a challenging task in image processing. Semi-blind deblurring is a useful technique which may be developed to restore the underlying sharp image given some assumed or known information about the cause of degradation. Existing models assume that the blur is of a particular type, such as Gaussian, and do not allow for the approximation of images corrupted by other blur types which are not easily incorporated into deblurring frameworks. We present an automated approach to image deconvolution which assumes that the cause of blur belongs to a set of common types. We develop a hierarchical approach with convolutional neural networks (CNNs) to distinguish between blur types, achieving an accuracy of 0.96 across a test set of 900 images, and to determine the blur strength, achieving accuracy of 0.77 across 1500 test images. Given this, we are able to reconstruct the underlying image to mean ISNR of 7.53.

Keywords: deconvolution, convolutional neural networks, colour fundus, retina, parametric

1 Introduction

Image deconvolution is a very useful tool amongst image preprocessing techniques which aims to remove blur which can hinder diagnosis with medical imaging and prevent further processing. In a current screening programme, approximately 5% of the images acquired are too blurred for assessment. It is also an important step for other techniques in image processing such as super resolution. While there exist models for image segmentation which can cope with some noise in an image, blur proves to be more of a problem for this as well as related tasks such as registration. Parametric kernel identification can be used to deblur images making some assumptions about the blur function. There exist

many models which work well without noise. However, noise is often present in images and can cause misidentification of the blur function and thus prevent accurate recovery of the image. This paper presents a model for restoring noisy, blurred images in which the blur is assumed to be of a certain type.

Assuming that the blur is spatially invariant, then denoting by h and u the blur function and true image respectively, we model the blurred image z as the convolution of the true image and kernel $z(x, y) = (h * u)(x, y) + \eta(x, y)$ where $*$ denotes the operation of convolution and η denotes noise. Deblurring (or deconvolution) [1] is the associated inverse problem which aims to recover the true image u from the received data z.

Deconvolution models may be split into 3 categories: non-blind deconvolution [2] involves recovering the true image with known blur function; blind deconvolution, [1] involves restoring the image with no knowledge of the blur function and is computationally expensive and difficult to achieve; semi-blind deblurring [3] involves recovering the hidden true image with only partial knowledge or assumptions about the blur function, such as the type of blur and offers a way of achieving an accelerated deblurring algorithm. Such models perform well and can obtain improved results over blind deblurring when the blur type may be known or estimated. Such techniques are useful in related areas such as the segmentation of blurred images and super-resolution where the blur is often of Gaussian or out of focus type. Our aim in this case is to recover the parameters and thus reconstruct the blur function.

Recently, deep neural networks (DNNs) have been emerged as a new area in machine learning analytics. Unlike conventional artificial neural networks, DNN layers are not fully connected and can learn to recognise complex nonlinear features. DNNs are also based use with graphical processing units (GPUs) which facilitate efficient training of large and complex machine learning techniques. Various deep learning architectures such as convolutional neural networks (CNNs) have been reported and developed for various applications including speech recognition [4] and bioinformatics [5] where they have been shown to produce state-of-the-art results on various tasks. Models for deconvolution employing machine learning approaches, including deep learning, have recently been reported. Schuler et al [6] developed a method for non-blind deblurring where the deconvolution procedure is learned with a multi-layer perceptron. The method is tested on synthetic examples and real out-of-focus images. The authors later extended this to the blind case for small blur kernels [7]. Xu et al. [8] also developed a neural network approach to non-blind deconvolution by linking to optimisation-based schemes, combining deconvolution with denoising. A semi-blind motion blur [9] approach to deconvolution was presented by Sun et al. [10] who used convolutional neural networks to estimate motion blur and adapted a uniform deconvolution approach to the non-uniform problem. In this paper, we develop a framework for semi-blind deblurring with a fast technique of finding the type and strength of blur from an image using CNNs which allow us to determine the blur function quickly and accurately.

2 Method

Our algorithm is separated into training the neural network to classify our data, and separate testing which includes classifying our images given the trained networks and producing a clean image. These stages can be done separately, so that the algorithm may be trained beforehand and later tested on individual examples without needing computationally-expensive training for each image, making this potentially a very fast approach to semi-blind image deconvolution.

For the first part of the training stage of our approach, we attempt to train a convolutional neural network to distinguish blur type using the training images with no assumptions on the blur; this includes the case of no blur being present. We also train individual networks to identify blur strength, for each of a set of blur types, given only the training images and an assumption on the type of blur; for example, we may assume that the blur is out-of-focus. At the testing stage, given an example image which may be blurred or not, we use the first trained network to determine whether blur is present in the image and, if so, the type of blur which is present. If the image is determined to not be blurred, nothing more is done and the image exits the algorithm. Since most images are likely to not contain blur, this means that most images can be processed in the approximate 0.0028 seconds that it takes the one CNN to classify it. If the image is deemed to blurred, the blur type should already be determined at the same time. Given this information, the appropriate pre-trained second CNN is used to classify the image by blur strength. This allows us to determine the kernel function. Since the kernel function is now assumed to be known, the problem is transformed into one of non-blind deconvolution which may be solved by an existing fast method such as [2]. This algorithm is presented in Figure 1 and the details of the CNN and deconvolution are shown below.

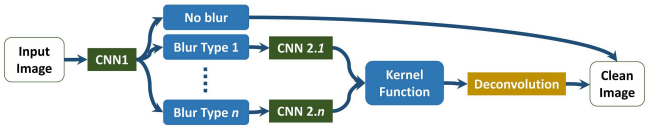


Fig. 1: Flow chart of our overall approach for testing an example image. Assuming that the CNNs are trained, the image is first classified as being corrupted by no blur or a particular type. If the image is blurred, a further CNN determines the strength of blur before the image is restored by a deconvolution process.

2.1 Classification using Convolutional Neural Networks

Convolutional neural networks (CNNs) are among the most popular deep neural network architectures and have achieved state-of-the-art results in image pattern recognition and other applications. CNNs learn features from raw data without the need for manually designing hand-crafted features.

A CNN is comprised of a modified version of LeNet convolutional neural network implemented by LeCun [11] with successive convolution layers alternated with subsampling and activation functions to implement the feature space for the input data. Fully connected layers combined with a softmax layer normalises the probability of the examples to be classified between 0 and 1. The convolution, nonlinearity, pooling, dropout and classification in a dense layer are the main operations in a convolutional neural network and considered the basic building blocks.The convolution layers work as feature detectors by extracting features from small squares of input image using filters of a certain window size allowing the pixels to preserve the spatial relationship between them. Moreover, to introduce the nonlinearity to convolutional neural networks, which typically use linear operations in the convolution operation, the Rectified Linear Unit (ReLU) is usually used as an activation function after each convolutional layer. To reduce the dimensionality of the feature maps resulting from the convolutional layers, a spatial sub-sampling (or pooling) layer is defined by sliding a square window on the input image and taking the maximum value in each small region. In addition to that, to decrease overfitting during the training stage, a dropout layer is used as a regularisation technique. The final layer is the conventional Multi Layer Perceptron which is a fully connected layer using fully connected nodes followed by a softmax activation function. The purpose of the dense layer (fully connected layer) is to use the features extracted from proceeding layers for classifying the input image into various labels based on the training examples.

The architecture of the CNN used in this experiment was structured as follows: four convolutional layers; each two followed by a non-linear activation function, a maxpooling layer, and dropout layer with dropping probability 0.25 and finally two fully connected layers; one has 512 neurons with dropout layer with dropping probability 0.5 and the other has 5 neurons represent the number of classes to be detected as shown in Figure 2.

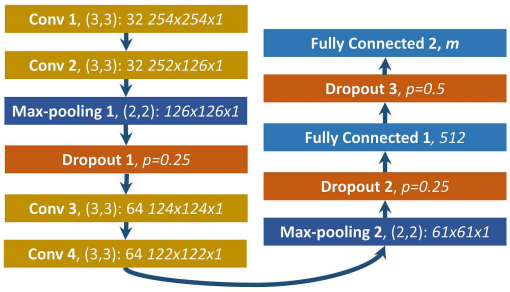


Fig. 2: Flow chart of our CNN architecture. m denotes the number of output classes.

The network was trained using stochastic gradient descent (SGD) with a constant learning rate of 0.001 and momentum parameter 0.9 by updating the

network weights which are initialised using Glorot weight initialisation. The objective function to be minimized was cross-entropy loss function $L = -tlog(p) - (1-t)log(1-p)$.

2.2 Non-Blind Image Deconvolution

Assuming that the blur kernel function is already determined, we aim to recover the sharp image by a deconvolution process. We might attempt to do this using a regularised variation approach which may be achieved by solving a minimising problem of an functional of the form

$$\min_{u(\mathbf{x})} \left\{ \|[h * u](\mathbf{x}) - z(\mathbf{x})\|^2_{L^2(\Omega)} + |\nabla u(\mathbf{x})| \right\} \tag{1}$$

where $h(\mathbf{x})$ denotes the blur function, $u(\mathbf{x})$ denotes the clean unknown image which we aim to recover, $z(\mathbf{x})$ denotes the blurred image which we started with, $\|\cdot\|^2_{L^2(\Omega)}$ denotes the L^2-norm over the domain and ∇ denotes the gradient. The first term in the energy functional is a fitting term which has a minima when $[h * u](\mathbf{x}) = z(\mathbf{x})$ and the second part is a regularisation term, in this case total variation, which aims to deal with the effects of noise which makes this problem ill-posed. While this can achieve good results, it can be quite slow to solve using traditional solvers such as gradient descent and conjugate gradient. The quality of results from this approach can also suffer from the relative simplicity of the model. Much work has been done recently in defining constrained deconvolution models which achieve improved results over more traditional ideas as well as fast solution techniques such split-Bregman and alternate direction methods. For this work, we reconstruct our images using an approach from Williams et al. [2] which provides implicitly constrained image deconvolution and a fast solver. The problem is presented as $\min_{u,\psi;\lambda}(f(u,\psi;\lambda))$ where

$$f(u,\psi;\lambda) = \frac{1}{2}||h * u - z||^2_{L^2(\Omega)} + \frac{\gamma}{2}||T_{\mathbf{a}}(\psi) - u||^2_{L^2(\Omega)} + < \lambda, T_{\mathbf{a}}(\psi) - u >$$

$$+\mu||\psi - \zeta||^2_{L^2(\Omega)} + \alpha \int_{\Omega} \left| \nabla \left(T_{\mathbf{a}}(\psi) + \theta||\psi - \zeta||^2_{L^2(\Omega)} \right) \right| \, \mathrm{d}\Omega, \tag{2}$$

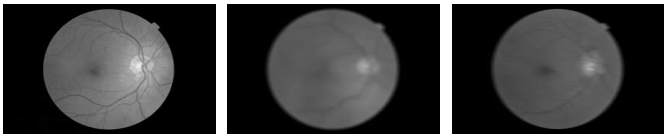
where the first term is a deconvolution fitting term, the second and third terms aim to constrain the values of the reconstructed image $u(\mathbf{x})$ to be similar to the implicitly constrained function $T(\psi)$ [2] using a lagrange multiplier λ. The fourth term is included to encourage convexity given certain parameters where ζ should be an initial approximation of the solution and $\psi(\mathbf{x})$ is a dual variable. The final component is a regularisation term. To minimise the functional, we first calculate ζ and proceed with alternate minimisation of the remaining arguments which can each be solved efficiently. More details are presented in [2].

3 Experimental Results

We test our approach using the Messidor dataset [12] of 1200 eye colour fundus images as our sharp, true data. These were acquired from patients with varying

stages of retinopathy or maculopathy at three ophthalmologic departments using a colour video 3CCD camera on a Topcon TRC NW6 non-mydriatic retinograph with a 45 degree field of view. The 8-bit images were captured at 1440×960, 2240×1488 or 2304×1536 pixels but, for our testing, we resize to 256×256. We split our datasets (to be defined in the results section) into training (75%) and testing (25%) sets; of the training data, 20% is reserved for validation. All tests were run on an HP Z440 Workstation with Intel Xeon E5-1620, 32GB RAM and an NVidia Titan Xp GPU which was used to train the CNNs.

We first aim to determine blur type. For this experiment, we consider the cases of no blur, out-of-focus blur and motion blur. We form a dataset for training and testing the CNNs composed of the 1200 messidor images which are sharp and uncorrupted; we consider these as class 1. We blur these images using out-of-focus blur of random diameter (strength) between 1 and 10 and consider these images as class 2. Similarly, we blur the class 1 sharp images using motion blur of random strength between 1 and 10; we consider these images as class 3. Examples are shown in Figure 3. From each class, 900 images are used for training, of which 180 are reserved for validation, and 300 images are used for testing once the network is trained. The network was trained until the cost function reached a plateau and then tested on the combined testing set. We achieved an accuracy of 0.9589. All of the clean images were classified correctly, while 99% of the out-of-focus images were classed correctly and 88.7% of the motion blurred images were classified correctly. The remaining incorrect images were all classified as clean.



(a) Clean Image (b) Out-of-focus Image (c) Motion Blurred Image

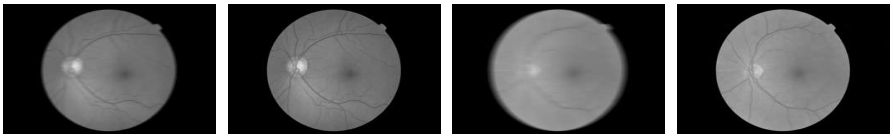
Fig. 3: Examples of training data: (a) clean image, (b) image corrupted by out-of-focus blur, (c) image corrupted by motion blur. In the cases of the blurred images, the strength of added blur corruption was random.

We now consider the feasibility of using neural networks to determine the strength of blur in an image with an obvious case which should be easily solvable. We define a set composed of the 1200 resized, clean Messidor images as class 0. We then blur these using a 20-pixel motion blur function and consider these as class 1. From each class, we randomly select 900 images for training, of which 180 are reserved for validation. The remaining 300 images from each class are combined into a test-set and are to be classified once the network is trained. The network was trained using the training and validation data to distinguish

between clean images and the heavily blurred images, and achieved an accuracy of 1.00 on the 600 test images, meaning that every image was classified correctly.

Given this encouraging result, we consider the same experiment but with a more difficult case using 10-tap motion blur. Training again until the relative cost was sufficiently small, we achieve 0.988 accuracy on 600 test images. Increasing the difficulty further by decreasing this to 3-tap blur, we are still able to achieve accuracy of 0.94. These results strongly suggest that may be able to use CNNs to reliably determine the unknown strength of blur in fundus images.

We consider a multi-class approach using a set of potential strength values including no blur, 4-, 6-, 8-, and 10-tap blur since less than 4 is not considered likely to considerably distort the image and greater than 10- is unlikely to happen in practice. To form a training and testing dataset, we blur the messidor dataset by each blur function and consider the resulting corrupted images as classified by blur strength. 75% of the images of each class (4500 images in total) are used for training, of which 20% (900 images) are reserved for validation. The remaining 1500 images are used for testing. We trained the network for 100 epochs. Using all five classes, we achieve 77% accuracy. Subsequent deconvolution resulted in a mean ISNR of 7.53 across 1200 examples which is similar to the 7.80 achieved by [3]. However, it should be noted that a different and smaller dataset was used for experiments. Some results are presenting in Figure 4.



(a) Blurred (b) Restored (c) Blurred (d) Restored

Fig. 4: Two examples of blurred images (a),(c) restored by our algorithm (c),(d).

4 Conclusion

We have presented a new technique for determining the type and level of blur in images, considering in particular colour fundus images of the retina. By developing convolutional neural networks for single and multi-class problems, we have been able to determine whether an image is blurred and how strong that blur is to a high accuracy. One considerable benefit is that the method naturally includes the same number of images in each class, thereby avoiding the issue of bias requiring solutions such as weighting. Building in image deconvolution, we create an automatic semi-blind deconvolution technique which does not require manual inspection of the images to determine or estimate the type and strength of blur present. An important aspect is to test this method beyond synthetic blur which requires creation of an annotated dataset. Further work could be considered in the future to improve the accuracy for distinguishing the strength of blur

so that the correct blur function may be identified, however the high accuracies found in this work are very encouraging. This approach may be extended to consider classification of other blur types which are not currently addressed. In particular, this approach is fast, automatic, and provides a technique for determining that an image is clean, estimating the blur and strength, and recovering the clean image from the blur degradation.

Acknowledgement

This project is funded by the National Institute for Health Researchs i4i Programme. This paper summarises independent research funded by the National Institute for Health Research (NIHR) under its i4i Programme (Grant Reference Number II-LA-0813-20005). B. Al-Bander acknowledges financial support from the Higher Committee for Education Development in Iraq (Grant No. 182).

References

1. T. F. Chan and C.-K. Wong, "Total variation blind deconvolution," *IEEE T. Image Process.*, vol. 7, no. 3, pp. 370–375, 1998.
2. B. M. Williams, K. Chen, and S. P. Harding, "A new constrained total variational deblurring model and its fast algorithm," *Numer. Algorithms*, vol. 69, no. 2, pp. 415–441, 2015.
3. M. S. C. Almeida and L. B. Almeida, "Blind and semi-blind deblurring of natural images," *IEEE T. Image Process.*, vol. 19, no. 1, pp. 36–52, 2010.
4. O. Abdel-Hamid, A.-r. Mohamed, H. Jiang, L. Deng, G. Penn, and D. Yu, "Convolutional neural networks for speech recognition," *IEEE/ACM TASLP*, vol. 22, no. 10, pp. 1533–1545, 2014.
5. T. Zeng, R. Li, R. Mukkamala, J. Ye, and S. Ji, "Deep convolutional neural networks for annotating gene expression patterns in the mouse brain," *BMC bioinformatics*, vol. 16, no. 1, p. 147, 2015.
6. C. J. Schuler, H. Christopher Burger, S. Harmeling, and B. Scholkopf, "A machine learning approach for non-blind image deconvolution," in *CVPR*, 2013, pp. 1067–1074.
7. C. J. Schuler, M. Hirsch, S. Harmeling, and B. Schölkopf, "Learning to deblur," *IEEE T. Pattern Anal.*, vol. 38, no. 7, pp. 1439–1451, 2016.
8. L. Xu, J. S. Ren, C. Liu, and J. Jia, "Deep convolutional neural network for image deconvolution," in *NIPS*, 2014, pp. 1790–1798.
9. A. Levin, "Blind motion deblurring using image statistics," in *Advances in Neural Information Processing Systems (NIPS)*, 2007.
10. J. Sun, W. Cao, Z. Xu, and J. Ponce, "Learning a convolutional neural network for non-uniform motion blur removal," in *CVPR*, 2015, pp. 769–777.
11. Y. LeCun, L. Jackel, L. Bottou, A. Brunot, C. Cortes, J. Denker, H. Drucker, I. Guyon, U. Muller, E. Sackinger *et al.*, "Comparison of learning algorithms for handwritten digit recognition," in *ICANN*, vol. 60, 1995, pp. 53–60.
12. E. Decencière, X. Zhang, G. Cazuguel, B. Laÿ, B. Cochener, C. Trone, P. Gain, R. Ordonez, P. Massin, A. Erginay *et al.*, "Feedback on a publicly distributed image database: the messidor database," *Image Anal. Stereology*, vol. 33, no. 3, pp. 231–234, 2014.